BAB 2
DASAR TEORI

2.1 KAJIAN PUSTAKA

Penelitian terkait dengan metode *HOG* dan *SVM* yang dilakukan oleh Intan Fatmawati [3] adalah pendeteksian kendaraan mobil untuk mendukung keamanan dalam berkendara agar tidak terjadi kecelakaan menggunakan metode serupa dan memakai kamera *Raspberry Pi* yang dipasang pada *dashboard* mobil untuk mendeteksi kendaraan yang berada di depan mobil tersebut. Di penelitian tersebut jika ada kendaraan yang jarak dari kendaraan kurang dari sama dengan 15 meter maka terdapat *buzzer* yang akan berbunyi untuk sebagai tanda jika ada kendaraan yang terlalu dekan dengan mobil tersebut sedangkan pada penelitian ini. Hasil akurasinya pada pengujian tersebut pada jarak 10 meter, 20 meter, dan 30 meter sebesar 81.3%, untuk pengujian akurasi dari integrase *hardware* dan *software* diperoleh 87.5%. Pada penelitian oleh Intan fatmawati yang menggunakan kamera *Raspberry Pi* yang dipasang di *dashboard* mobil sedangkan untuk penelitian yang akan penulis lakukan adalah dengan menggunakan kamera dari *ATCS* Dinas perhubungan Purwokerto.

Penelitian yang dilakukan oleh Alvin Lazaro [4] mendeteksi sebuah kendaraan di jalan menggunakan *OpenCV*, pada penelitian ini mendeteksi jenis suatu kendaraan dan menghitung kendaraan yang terdeteksi yang berdasarkan jenis kendaraan tersebut. Inputan dari penelitian ini berupaya video dengan memasukkan video rekaman pada lalu lintas ke program yang menghasilkan sebuah file .txt sebagai *ouputannya.* Hasil pengujian pada penelitian ini yang memiliki akurasi sebesar 77.8% untuk kondisi jalanan yang sepi, 47.5% untuk kondisi jalan yang normal, dan 28.2% untuk jalanan yang padat akan kendaraan. Pada penelitian yang dilakukan oleh Alvin Lazaro menggunakan metode *Haar-Like Feature* sedangkan peneltian yang akan penulis teliti menggunakan metode *HOG* dan *SVM*.

Berdasarkan pada penelitian yang dilakukan oleh Cahyo permata [5] pada penelitian ini menggunakan metode *HOG* yang dapat mendeteksi mobil.

Penelitiaan ini dilakukan dengan 2462 citra positif untuk sebuah mobil, dan 4627 untuk citra negatif. Pengujian pada penelitian ini dilakukan dengan mengubah suatu jumlah dari data training dan testing dengan mengubah nilai dari threshold pada area yang dideteksi. Akurasi dari pengujian ini sebesar 99.10% dengan data citra sebanyak 1028. Untuk hasil pendeteksian mobil hasilnya adalah 76.17% dengan data citra sebanyak 196. Pada penelitian Cahyo permata hanya mendeteksi satu jenis mobil saja, sedangkan untuk penelitian yang akan penulis kembangkan adalah dengan mendeteksi mobil dan becak.

Pada penelitian yang dilakukan Irawan [1] yang mendeteksi mobil dengan menggunakan metode ekstrasi fitur HOG dan metode klasifikasi menggunakan *Support Vector Machine (SVM)* Pengujian yang dilakukan pada 100 citra dengan ukuran 640x480 pixel didapatkan tingkat akurasi rata-rata sebesar 97,25%. Dan pada Penelitian Wang [2] pendekatan dua langkah untuk deteksi kendaraan diusulkan. Langkah pertama dari Pendekatannya adalah untuk memperkirakan lokasi potensial kendaraan melalui penelusuran area bayangan kendaraan bagian rendah. Untuk menemukan bayangan ini, fitur mirip Haar dengan Adaboost digunakan untuk melatih detektor Haar secara offline dan proses pembelajaran kembali dengan sampel pelatihan keras diterapkan untuk meningkatkan tingkat deteksi. Berdasarkan pemrosesan sebelumnya, ROI (Region of interest) + Algoritma HOG + SVM digunakan untuk verifikasi kendaraan. Akhirnya, pendekatan K-means digunakan untuk menggabungkan hasil deteksi yang serupa. Hasil percobaan membuktikan bahwa sistem kami dapat digunakan untuk pendeteksian kendaraan sebelumnya secara real-time dengan kuat dan akurat.

Tabel 2. 1 Rangkuman Studi Pustaka

No	Peneliti	Judul	Metode	Hasil
1.	Frans Irawan 2015	Deteksi Mobil pada Citra Digital Menggunakan C-HOG dan Support Vector Machine	C-HOG dan SVM	Tingkat akurasi rata-rata yang dicapai untuk mendeteksi mobil pada citra

				ialah sebesar 97,25%.
2.	Huan Wang dan Haichuan Zhang 2014	A hybrid method of vehicle detection based on computer vision for intelligent transportation system	ROI (Region of interest) + HOG + SVM	Tingkat deteksi tertinggi yang dicapai ialah 92,73%.
3.	Intan Fatmawati 2019	Deteksi Kendaraan Roda Empat Untuk Mendukung Keamanan Berkendara Menggunakan Histogram of Oriented Gradients dan Support Vector Machine Berbasis Raspberry Pi	*HOG* dan *SVM*	Hasil akurasi dengan pengujian pada jarak 10 m, 15 m, 20 m, 30 m sebesar 81.3%, untuk pengujian akurasi integrasi Hardware dan Software sebesar 87.5%.
4	Alvin Lazaro 2017	Deteksi Jenis Kendaraan Di Jalan Menggunakan Opencv	Haar-like Feature	Hasil pengujiannya didapatkan nilai akurasi rata-rata 77.8% untuk kondisi jalan sepi, 47.5% untuk kondisi jalan normal, dan 28.2%

				untuk kondisi jalan padat.
5	Cahyo Permata 2012	Deteksi Mobil Menggunakan Histogram of Oriented Gradient	*HOG* dan *SVM*	Tingkat akurasi tes model dengan image tes sebesar 99,10% dengan image tes sebanyak 1028 image positif dan 1978 image negatif. Sedangkan hasil deteksi mobil menunjukkan 76,17% dari 196 image.

2.2 DASAR TEORI

2.2.1 Sistem Otomasi Klasifikasi Objek Berdasarkan Citra

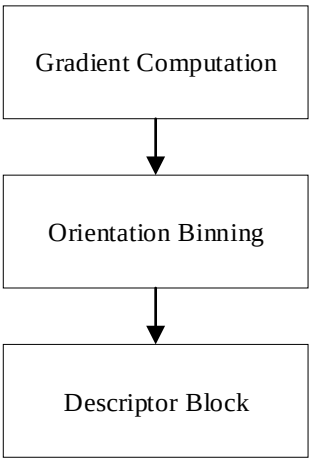
Otomasi adalah sebuah ilmu yang mana kita dituntut untuk membuat ataupun merubah sebuah mesin yang dijalankan manual menjadi otomatis.Pada dasarnya otomasi ini digunakan untuk membantu manusia dalam melakukan hal-hal rutin, keterbatasan dalam hal ketelitian adalah hal yang dimiliki oleh manusia, berbeda dengan mesin ataupun computer dengan otomasi, manusia dapat terbantu atau meringankan pekerjaannya.

Alasan mengapa otomasi adalah, karena kita hidup dijaman yang serba canggih dan modern, dimana semuanya dapat dikerjakan dengan otomatis, selain dapat meringankan perkerjaan, otomasi dapat menghemat biaya waktu dan tenaga, sehingga dapat mempercepat perkembangan dalam dunia industry itu sendiri.

2.2.2 Sistem Ekstrasi Menggunakan *Histogram of Oriented Gradient (HOG)*

Histogram of Oriented Gradient (HOG) adalah salah satu metode ekstrasi fitur pada pengolahan citra yang dapat mendeteksi objek. Suatu citra dapat diperoleh informasi dan akan dibagi menjadi sebuah *cell*, lalu tiap *cell* akan dihitug sebagai *histogram of oriented gradient.* Pada tiap *pixel* dalam *cell* berperan pada saat voting bobot dalam membagun histogram yang orientasinya pada nilai gradient yang dihitung [6]

Hal terpenting dari HOG ialah tampilan lokal obyek dan bentuk dalam sebuah citra dapat dijelaskan oleh distribusi intensitas gradien atau arah tepi. Implementasi HOG dapat dicapai dengan membagi *citra* menjadi suatu bagian yang kecil dan saling terhubung yang dapat disebut sebagai sel, dan untuk setiap sel menyusun gradien arah atau orientasi tepi untuk *pixels*. Untuk peningkatan kinerja, histogram lokal dapat dinormalisasi secara kontras dengan menghitung ukuran intensitas pada seluruh wilayah besar dari citra yang disebut blok, kemudian menggunakan nilai ini untuk normalisasi semua sel dalam blok[6].



Gambar 2.1 Implementasi dari *HOG* [6]

Pada Gambar 2.2, adalah langkah-langkah implementasi dari HOG dimulai dari *Gradient Computation, Orientation Binning*, dan *Descriptor Block* atau *Block Normalization*. Pada masing-masing langkah harus ada fungsi yang dicari.

Langkah pertama dari implementasi HOG adalah menghitung nilai gradien dari citra yang telah ditentukan *filtering* citra *grayscale* dengan menggunakan operasi *central differences* sebagai berikut: [7]

$$Ix(r,c) = \frac{I(r,c+1)-I(r,c-1)}{2} \quad 2.1$$

$$Iy(r,c) = \frac{I(r+1,c)-I(r-1,c)}{2} \quad 2.2$$

Dimana r merupakan baris dari matriks dan c merupakan kolom dari matriks. Kemudian mencari besar dari gradien (μ) dan orientasi dari gradien (θ) dengan operasi sebagai berikut.

$$\mu = \sqrt{Ix^2 + Iy^2} \quad [7] \quad 2.3$$

$$x = \frac{180}{\pi} \ (tam_{2^{-1}}(Iy, Ix) mod \pi) \quad 2.4$$

Langkah selanjutnya adalah pembuatan *cell* pada citra serta pencarian nilai-nilai dari *bin* pada setiap *cell*. Bentuk dari *cell* mempengaruhi *pixel* yang diambil dimana bentuk *cell* ditentukan dari tipe geometri blok ditahap pembentukan sebuah blok. Pada setiap *pixel* didalam *cell* mempunyai nilai bin yang dapat dicari dengan menggunakan operasi sebagai berikut.

$$V_j = \mu \frac{c_{j+1}-\theta}{w} \text{ pada bin } j = \left[\frac{\theta}{w} - \frac{1}{2}\right] mod\ B \quad [7][7]$$

2.5

$$V_{j+1} = \mu \frac{\theta - cj}{w} \text{ pada bin } (j+1)\ mod\ B \quad 2.6$$

Dimana V_j ialah nilai dari bin *j*, B adalah *bin* yang digunakan, w merupakan jarak dari bin yaitu $\frac{180}{B}$ dan $w\ (j + \frac{1}{2})$. Nilai *bin* yang digunakan adalah 9, sehingga =180/9=20.

Langkah terakhir adalah untuk memperhitungkan perubahan pada pencahayaan dan kontras, kekuatan dari *gradien* harus dinormalisasi secara lokal, dan memerlukan pengelompokan *cell* menjadi sebuah blok yang besar dan dapat terhubung [5]

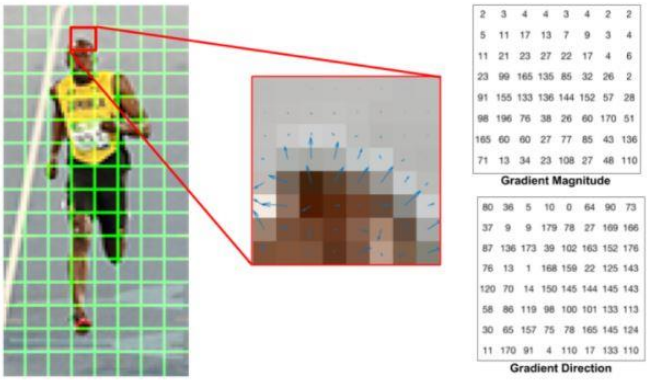
a. **Menghitung HOG 8x8 *cell***

Salah satu alasan penting menggunakan deskriptor fitur untuk mendeskripsikan patch gambar adalah karena deskriptor fitur menyediakan representasi yang ringkas. Patch gambar 8 × 8 berisi nilai 8x8x3 = 192 piksel. Gradien patch ini berisi 2 nilai (besaran dan arah)

per piksel yang berjumlah 8x8x2 = 128 angka. Di akhir bagian ini akan melihat bagaimana 128 angka ini direpresentasikan menggunakan histogram 9-bin yang dapat disimpan sebagai larik 9 angka. Tidak hanya representasi yang lebih ringkas, menghitung histogram melalui tambalan membuat representasi ini lebih kuat terhadap noise. Graident individu mungkin memiliki noise, tetapi histogram lebih dari 8 × 8 patch membuat representasi tersebut kurang sensitif terhadap noise.

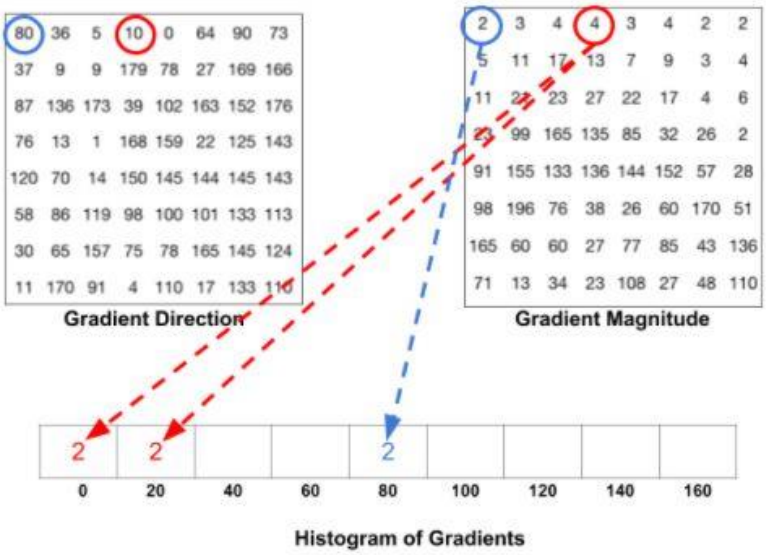
Sel 8 × 8 dalam foto pejalan kaki dengan skala 64 × 128 cukup besar untuk menangkap fitur-fitur menarik (misalnya wajah, bagian atas kepala, dll.).

Histogram pada dasarnya adalah vektor (atau larik) dari 9 bin (angka) yang sesuai dengan sudut 0, 20, 40, 60… 160.[8]



Gambar 2.2 HOG

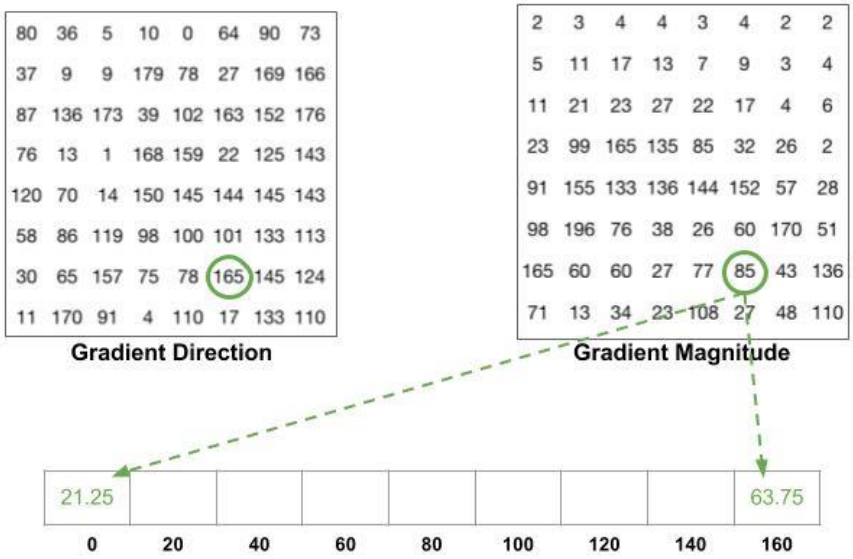
Yang mewakili gradien dalam sel 8 × 8 dengan satu perbedaan kecil - sudutnya antara 0 dan 180 derajat. Ini disebut gradien "unsigned" karena gradien dan negatif diwakili oleh angka yang sama.



Gambar 2.3 HOG

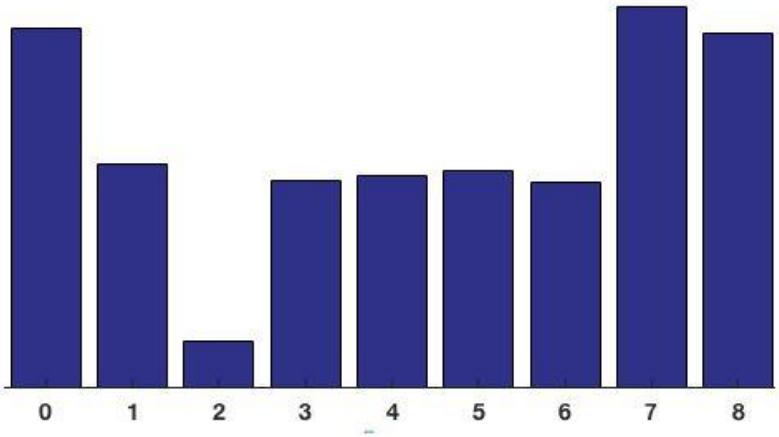
Langkah selanjutnya adalah membuat histogram gradien di sel 8 × 8 ini. Histogram berisi 9 bin yang sesuai dengan sudut 0, 20, 40… 160.

Gambar berikut mengilustrasikan prosesnya. Kami melihat besarnya dan arah gradien dari patch 8 × 8 yang sama seperti pada gambar sebelumnya. Sebuah bin dipilih berdasarkan arah, dan suara (nilai yang masuk ke dalam bin) dipilih berdasarkan besarnya. Pertama mari kita fokus pada piksel yang dilingkari dengan warna biru. Ini memiliki sudut (arah) 80 derajat dan besarnya 2. Jadi itu menambahkan 2 ke bin ke-5. Gradien pada piksel yang dilingkari menggunakan warna merah memiliki sudut 10 derajat dan besarnya 4. Karena 10 derajat adalah setengah jalan antara 0 dan 20, *vote* oleh piksel akan terbagi secara merata ke dalam dua bins.



Gambar 2.4 HOG

Kontribusi dari semua piksel dalam 8X8 ditambahkan untuk membuat histogram 9 bin. Dapat dilihat pada gambar 2.5



Gambar 2.5 Histogram

Dalam representasi kami, sumbu y adalah 0 derajat. Anda dapat melihat histogram memiliki banyak bobot mendekati 0 dan 180 derajat, yang merupakan cara lain untuk mengatakan bahwa dalam *gradien patch* menunjuk ke atas atau ke bawah.

2.2.3 Sistem Klasifikasi Objek menggunakan *Support Vector Machine (SVM)*

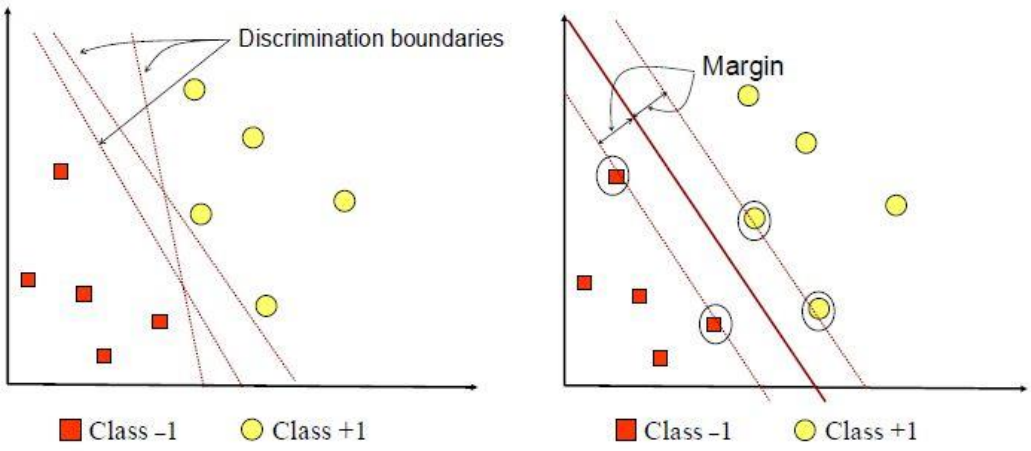
Support Vector Machine adalah sebuah algoritma Supervised untuk memisahkan dua buah *class* didalam ruang dimensi. *SVM* memisahkan *hyperplane* dengan *margin* yang terbesar, dimana *margin* adalah jarak dari pemisah *hyperplane* ke *training* terdekat[9]. Cara kerja dari SVM ini ada 3 yaitu :

1. Support Vector

   Data yang memiliki jarak yang terdekat yang berasal dari sebuah kelompok yang berbeda.
2. Hyperlane

   Garis pembats antara 2 *support vector. Hyperlane* ini sediri adalah sebuah garis lurus atau bidang mendatar yang memisahkan kelas-kelas.
3. Margin

   Jarak *support vector* dengan *hyperlane*



Gambar 2.6 *SVM* [2]

Gambar diatas adalah menunjukan ada memperlihatkan beberapa pattern yang termasuk anggota dari dua buah class : +1 dan -1. Pattern yang tergabung pada class -1 disimbolkan dengan warna merah, sedangkan pattern pada class +1, disimbolkan dengan warna kuning. Problem klasifikasi dapat diterjemahkan dengan usaha menemukan garis (hyperplane) yang memisahkan antara kedua kelompok tersebut. Berbagai alternatif garis

pemisah (discrimination boundaries) ditunjukkan pada gambar 2.6. Hyperplane adalah pemisah terbaik antara kedua class dapat ditemukan dengan mengukur margin hyperplane tersebut dan mencari titik maksimalnya. Margin adalah jarak antara hyperplane tersebut dengan pattern terdekat dari masing-masing class. Pattern yang paling dekat ini disebut sebagai *support vector*. Garis solid pada gambar 2.6 menunjukkan hyperplane yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua class, sedangkan titik merah dan kuning yang berada dalam lingkaran hitam adalah support vector. Usaha untuk mencari lokasi hyperplane ini merupakan inti dari proses pembelajaran pada SVM.

Karakteristik dari SVM adalah, prinsip kerja pada SVM pada dasarnya hanya mampu menangani klasifikasi 2 buah class, SVM adalah linear classifier, Pattern recognition dilakukan dengan mentransformasikan data pada input space ke ruang yang berdimensi lebih tinggi, dan optimisasi dilakukan pada ruang vector yang baru tersebut. Hal ini membedakan SVM dari solusi pattern recognition pada umumnya, yang melakukan optimisasi parameter pada ruang hasil transformasi yang berdimensi lebih rendah daripada dimensi input space.

2.2.4 **Parameter Kinerja Sistem Klasifikasi**

Confusion matrix adalah sebuah alat yang digunakan untuk mengevaluasi sebuah model klasifikasi yang melih objek yang dideteksi benar atau salah. Terdapat beberapa ketentuan dalam menjalankan *confusion matrix* ini ada *True positive, False Positive, True Negative,* dan *False Negative* [4].

Tabel 2. 2 Confusion Matriks

	Predicted = Yes	*Predicted = No*
Actual = Yes	*True Positive*	*False Negative*
Actual = No	*False Positive*	*True Negative*

Pada table 2.2.7 diatas, *true positive* (TP) adalah sejumlah data atau citra positif yang diklasifikasikan sebagai positif, *true negative* (TN) adalah

citra negatif yang diklasifikasikan negatif, *false positive* (FP) adalah sejumlah data citra negatif yang diklasifikasikan sebagai positif, dan *false negative (FN)* adalah sejumlah data citra positif yang diklasifikasikan sebagai negative. Setelah mendapatkan hasil citra yang telah diuji lalu akan didapatkan hasil dari *confusion matrix* sehingga jumllah akurasi, spesifikasi dan sensitivitas dapat dihitung [4].

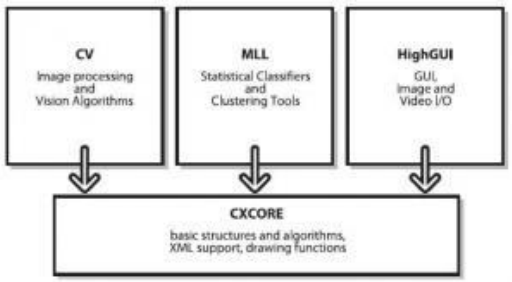
2.2.5 Perangkat Klasifikasi Objek Menggunakan Citra Berbasis *OpenCV* Dan *Python*

1) OpenCV

Open Computer Vision (OpenCV) adalah sebuah *Application Programming Interface (API) library* pada pengolahan citra atau *computer vision*. Visi computer itu sendiri adalah suatu bidang dari pengolahan citra yang dapat melihat seperti manusia. *OpenCV* termasuk kedalam *library open source*, yang memiliki fungsi yang baik untuk *image/video.* [10]

OpenCV terdapat 5 library, diantaranya adalah:

1. CV : untuk algoritma Image processing dan Vision.
2. ML : untuk machine learning library
3. Highgui : untuk GUI, Image dan Video I/O.
4. CXCORE : untuk struktur data, support XML dan fungsi-fungsi grafis.
5. CvAux : untuk struktur dan Konten OpenCV



Gambar 2.7 Struktur dan konter OpenCV [9]

2) Python

Python merupakan salah satu Bahasa pemorograman yang berorientasi *object*. Bahasa pemrograman python dapat berjalan pada berbagai sistem operasi lainnya. Beberapa kelebihan yang dimiliki Bahasa pemorgaram python adalah memiliki *library* yang luas dan tersedia beberapa modul-modul, memiliki aturan layout yang dapat memudahkan untuk pengecekan penulisan, *orientation object*, tata Bahasa yang mudah untuk dipelajari

Contoh library *pyhton* adalah :

1. Matplotlib

   Matplotlib merupakan library didalam python yang digunakan untuk memproses gambar-gambar, baik 2 dimensi maupun 3 dimensi.

2. Numpy

   Numpy *(Numerical Python)* merupakan *library add-on python* yang fungsinya untuk memproses data numerik.